Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 1 (2012) 1635–1644

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2010

On the robustness of a one-period look-ahead policy in multi-armed bandit problems

Ilya O. Ryzhov[a], Peter I. Frazier[b], Warren B. Powell[a]

[a]*Dept. of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08540, USA*

[b]*Dept. of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853, USA*

Abstract

We analyze the robustness of a knowledge gradient (KG) policy for the multi-armed bandit problem. The KG policy is based on a one-period look-ahead, which is known to underperform in other learning problems when the marginal value of information is non-concave. We present an adjustment that corrects for non-concavity and approximates a multi-step look-ahead, and compare its performance to the unadjusted KG policy and other heuristics. We provide guidance for determining when adjustment will improve performance, and when it is unnecessary. We present evidence suggesting that KG is generally robust in the multi-armed bandit setting, which argues in favour of KG as an alternative to index policies.

Keywords: multi-armed bandit, knowledge gradient, optimal learning, Bayesian learning

1. Introduction

The multi-armed bandit problem is a classic problem in sequential analysis. Suppose there are M independent reward processes whose means are stationary but unknown. We activate one reward process at a time, and collect a random payoff. For example, the processes could be the rewards obtained by playing different slot machines. In addition to its immediate value, the payoff obtained by pulling the arm of a slot machine allows us to construct a better estimate of the mean reward of that particular machine. The objective is to maximize the total reward collected across N plays.

The bandit problem provides an elegant illustration of the exploration vs. exploitation dilemma. When choosing between reward processes, we must strike a balance between choosing a process about which we are uncertain to see whether it has a high reward (exploration), and choosing a process that we believe to have a high reward (exploitation). This dilemma arises in numerous application areas. Examples include:

1. *Clinical trials*. We are testing experimental drug treatments on human patients. Each arm represents the effectiveness of a particular treatment. We wish to find the most effective treatment, while being mindful of the

Email address: iryzhov@princeton.edu (Ilya O. Ryzhov)

1877-0509
doi:10.1016/j.procs.2010.04.183

outcomes of individual trials. Much of the classic work on bandit problems has been motivated by the problem of clinical trials [1, 2].

2. *E-commerce*. An online advertising system can choose one advertisement to display at a given time. Each advertisement attracts a certain number of clicks, generating revenue for the system. We wish to find the most profitable advertisement, while maximizing the total value obtained across all advertisements tested. This application is considered in [3].
3. *Energy portfolio selection*. Certain new technologies have the potential to reduce greenhouse gas emissions in residential households (e.g. improved ventilation, energy-efficient appliances, solar panels). We can install a portfolio of several technologies into a residential building to observe its effectiveness. We wish to find the most energy-efficient portfolio, but we also would like to ensure a good outcome for every building that we refit. This application is discussed in [4].

Many applications go beyond the standard multi-armed bandit setting, laid out in [5, 6]. However, the multi-armed bandit model provides a clean and elegant mathematical framework for reasoning about the issue of exploration vs. exploitation, an issue that arises in every one of the problems listed above. For this reason, the bandit setting provides insight into complicated applications, and has attracted a great deal of attention in the literature.

A key advance in the bandit literature was the development of index policies. In every time step, an index policy computes an index for every arm, then pulls the arm with the highest index. The index of an arm depends on our estimate of the reward of that arm, but not on our estimates of other rewards. Thus, an index policy decomposes the problem, and considers every arm separately, as if that arm were the only arm in the problem. Most well-known algorithms for bandit problems are index policies, including interval estimation [7], upper confidence bounding [8, 9], and the Gittins index policy of [1, 10]. In particular, the Gittins index policy is asymptotically optimal as $N \to \infty$ when the objective function is discounted. However, Gittins indices are difficult to compute, giving rise to a body of work on approximating them [11, 12, 13].

A more recent approach is the method of knowledge gradients. Originally, this method was developed by [14] for the ranking and selection problem, an offline learning problem where the objective is simply to find the arm with the highest reward, not to maximize the total reward collected over N time steps. The knowledge gradient (KG) algorithm was also studied by [15, 16, 17] in the context of ranking and selection. The KG method chooses the arm that is expected to make the greatest single-period improvement in our estimate of the best mean reward. In this way, KG looks ahead one time step into the future and considers the way in which our estimates will change as a result of pulling a particular arm. The algorithm is thus optimal for $N = 1$, since it computes the value of a single pull exactly. In many offline settings, it is optimal as $N \to \infty$ as well.

The KG method was extended to the multi-armed bandit setting in [18, 19]. While the KG method is suboptimal, it does not require the difficult calculations necessary to compute Gittins indices, and can often outperform Gittins approximations in practice [4, 19]. However, the robustness of this approach remains an important topic for study. Because KG only considers the value of pulling an arm one time, it is important to consider if there is some additional benefit in looking ahead more than one time step. The work by [20] examines this question in the ranking and selection problem, and finds that the KG method can underperform when the marginal value of information is non-concave in the number of times an arm is pulled.

In this paper, we test the robustness of the KG method in the bandit setting. We consider an adjusted policy that approximates the value of information over multiple time steps. The adjusted policy performs similarly to the original online KG policy of [18], but offers substantial improvement in a restricted class of problems where N is large. Both adjusted and original KG consistently outperform a number of leading index policies. Our results provide important evidence in support of the viability of the KG approach as a robust alternative to index policies.

2. The multi-armed bandit problem

Suppose that there are M arms or reward processes. Let μ_x be the unknown mean reward of arm x. By pulling arm x, we receive a random reward $\hat{\mu}_x \sim \mathcal{N}\left(\mu_x, \lambda_x^2\right)$, where the *measurement noise* λ_x^2 is a known constant. Though μ_x is unknown, we assume that $\mu_x \sim \mathcal{N}\left(\mu_x^0, \left(\sigma_x^0\right)^2\right)$. Thus, our distribution of belief on μ_x is encoded by the pair $\left(\mu_x^0, \sigma_x^0\right)$. The random payoffs and mean rewards of different arms are assumed to be independent.

We say that something occurs "at time n" if it happens after n arm pulls, but before the $(n+1)$st. Let $x^n \in \{1, ..., M\}$ be the $(n+1)$st arm we pull. Then, $\hat{\mu}_{x^n}^{n+1}$ is the random reward collected as a result of the $(n+1)$st pull. For each x, the time-n posterior distribution of belief on μ_x is $\mathcal{N}\left(\mu_x^n, (\sigma_x^n)^2\right)$. If we measure x^n at time n, the posterior distributions change as follows:

$$\mu_x^{n+1} = \begin{cases} \frac{(\sigma_x^n)^{-2}\mu_x^n + \lambda_x^{-2}\hat{\mu}_x^{n+1}}{(\sigma_x^n)^{-2} + \lambda_x^{-2}} & x = x^n \\ \mu_x^n & x \neq x^n \end{cases} \tag{1}$$

Because the rewards are believed to be independent, only one set of beliefs is updated per time step. The variance of our beliefs is updated as follows:

$$\left(\sigma_x^{n+1}\right)^2 = \begin{cases} \left[(\sigma_x^n)^{-2} + \lambda_x^{-2}\right]^{-1} & x = x^n \\ (\sigma_x^n)^2 & x \neq x^n \end{cases} \tag{2}$$

The derivation of these Bayesian updating equations is a simple application of Bayes' rule, and can be found in [21]. If we let $\mu^n = \left\{\mu_1^n, ..., \mu_M^n\right\}$ and $\sigma^n = \left\{\sigma_1^n, ..., \sigma_M^n\right\}$, then the *knowledge state* $s^n = (\mu^n, \sigma^n)$ parameterizes all of our beliefs, and (1-2) describe the evolution of s^n into s^{n+1}.

Observe that μ_x^{n+1} becomes known at time $n+1$, but is random from the point of view at time n. Suppose now that $x^n = x^{n+1} = ... = x^{n+m-1}$, that is, we pull the same arm m times, starting at time n. It can be shown that the conditional distribution of μ_x^{n+m} given s^n and given $x^n = ... = x^{n+m-1}$ is $\mathcal{N}\left(\mu_x^n, \tilde{\sigma}_x^n(m)^2\right)$, where

$$\tilde{\sigma}_x^n(m)^2 = \frac{(\sigma_x^n)^2 m}{\left(\lambda_x^2/(\sigma_x^n)^2\right) + m} \tag{3}$$

by the conditional variance formula. For more details, see [21]. We will find this fact useful in our discussion of the KG policy.

It remains to define the objective function. Suppose that we are allowed to pull N arms. Our goal is to choose $x^0, ..., x^{N-1}$ to maximize the total expected reward that we collect. To allow ourselves to make decisions adaptively as our beliefs evolve, we define a *policy* π to be a sequence of *decision rules* $X^{\pi,0}, ..., X^{\pi,N-1}$. Each decision rule $X^{\pi,n}$ is a function mapping the knowledge state s^n to an element of the set $\{1, ..., M\}$, thus telling us which arm to pull at time n, based on the knowledge state s^n. Our objective can be stated as

$$\sup_\pi \mathbb{E}^\pi \sum_{n=0}^{N-1} \mu_{X^{\pi,n}(s^n)}, \tag{4}$$

where $\mathbb{E}^\pi$ is an expectation over the outcomes of all arm pulls, given that they are chosen in accordance with the policy π. Some studies of bandit problems, such as [22], restate (4) as

$$\inf_\pi \mathbb{E}^\pi \sum_{n=0}^{N-1} \max_x \mu_x - \mu_{X^{\pi,n}(s^n)}.$$

This is known as "regret minimization," and addresses the same goal of maximizing the total reward collected.

3. Using knowledge gradients in bandit problems

The online KG policy set forth in [18, 19] chooses an arm at time n using the decision rule

$$X^{KG,n}(s^n) = \arg\max_x \mu_x^n + (N-n-1)\,\mathbb{E}_x^n\left(\max_{x'} \mu_{x'}^{n+1} - \max_{x'} \mu_{x'}^n\right), \tag{5}$$

where $\mathbb{E}_x^n$ is an expectation given $x^n = x$ and s^n. The difference inside this expectation is the improvement that our time-$(n+1)$ estimate $\max_{x'} \mu_{x'}^{n+1}$ of the best reward makes over our time-n estimate $\max_{x'} \mu_{x'}^n$. The closed-form solution of this expectation follows from the next result.

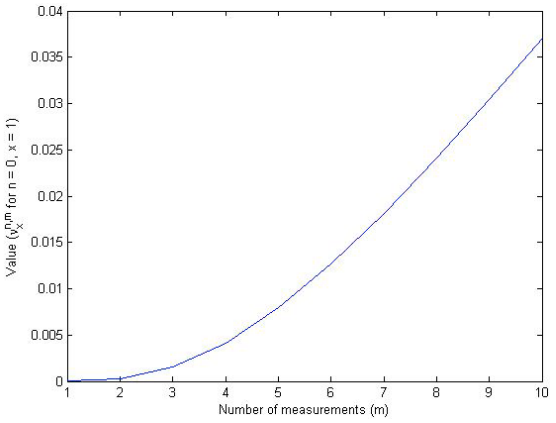


Figure 1: The value $\nu_1^{0,m}$ as a function of m for the two-armed example where arm 0 has known value 0, arm 1 is described by $\mu_1^0 = -1$ and $\left(\sigma_1^0\right)^2 = 25$, and $\lambda^2 = 10^4$.

Lemma 1. *Define*

$$\nu_x^{n,m} = \mathbf{E}^n \left(\max_{x'} \mu_{x'}^{n+m} \mid x^n = ... = x^{n+m-1} = x \right) - \max_{x'} \mu_{x'}^n$$

to be the expected improvement made by pulling arm x exactly m times in a row, starting at time n. Then,

$$\nu_x^{n,m} = \tilde{\sigma}_x^n(m) \, f\left(-\frac{\left|\mu_x^n - \max_{x' \neq x} \mu_{x'}^n\right|}{\tilde{\sigma}_x^n(m)} \right) \tag{6}$$

where $f(z) = z\Phi(z) + \phi(z)$ and Φ, ϕ are the standard Gaussian cdf and pdf.

This result can be shown using the analysis of [15], with $\tilde{\sigma}_x^n(m)$ replacing $\tilde{\sigma}_x^n(1)$ throughout. Using Lemma 1, we can easily write the KG decision rule as

$$X^{KG,n}(s^n) = \arg\max_x \mu_x^n + (N - n - 1)\, \nu_x^{n,1}. \tag{7}$$

The online KG policy approximates the value of pulling an arm x with the sum of the immediate expected reward, μ_x^n, and an "exploration bonus," $(N - n - 1)\nu_x^{n,1}$. The quantity $\nu_x^{n,1}$ represents the extra benefit per time period obtained from pulling x once. From (6), it is clear that KG is not an index policy, because $\nu_x^{n,1}$ depends on $\max_{x' \neq x} \mu_{x'}^n$ as well as on μ_x^n. Note that [18, 19] use $N - n$ instead of $N - n - 1$ when writing (5) and (7), because the models in these studies allow one last pull at time N.

In some situations, the quantity $\nu_x^{n,1}$ undervalues the true per-time-period value of the knowledge gained by pulling x. The KG factor $\nu_x^{n,1}$ is computed under the assumption that no future observations will be made, but in fact observations will be made in the future. Furthermore, the value of one piece of information often depends critically on what other information can be collected [23, 24, 25]. Several pieces of information can have little or no value on their own, but when combined together can have substantial value. In our case, the value $\nu_x^{n,m}$ of m pulls is not concave in m.

To illustrate the issue, consider a simple problem with two arms, with arm 0 having known value 0. Our prior on the value of arm 1 is $\mathcal{N}(-1, 25)$. The measurement noise has variance $\lambda^2 = 10^4$ and the horizon is $N = 10^5$. The online KG factor for arm 0 is 0. For the unknown arm, applying (6) yields $\nu_1^{0,1} = 1.7 \times 10^{-6}$, a very small number. As a result, the KG policy pulls the known arm. Since pulling the known arm does not provide any information and leaves the posterior equal to the prior, the online KG policy always pulls arm 0, getting a total reward of 0. The non-concavity of $\nu_1^{0,m}$ can be clearly seen in Figure 1.

Although the value $\nu_1^{0,1}$ of a single pull of the unknown arm is miniscule, the value of 10 measurements, $\nu_1^{0,10} = 0.037$ is reasonably large. Compare the KG policy to the simplistic policy that pulls the unknown arm 10 times,

then pulls the arm that appears to be the best until the end of the horizon. This simplistic policy has expected value $10\mu_1^0 + (N-10)\nu_1^{0,10} = 3690$, much greater than the KG policy's value of 0. This poor performance is caused by non-concavity in the value of information, but it can be fixed by the adjustment described in the next section.

4. Adjusting the knowledge gradient policy

We propose an adjusted version of KG, referred to as KG(*). The policy is derived by considering the value obtained by pulling an arm x several times, then afterward selecting the arm with the largest posterior mean and pulling it until the end of the horizon. This derivation is analogous to the logic used to adjust KG for ranking and selection in [20, 26].

Let $\mu_*^n = \max_x \mu_x^n$ for notational convenience. Suppose that we pull arm x exactly m times in a row. Then, the expected value of the reward obtained is

$$m\mu_x^n + (N-n-m)(\nu_x^{n,m} + \mu_*^n) \tag{8}$$

where $m\mu_x^n$ is the immediate expected reward obtained from the first m samples, all from arm x, and $(N-n-m)(\nu_x^{n,m}+\mu_*^n)$ is the expected reward obtained by pulling (until the end of the horizon) the arm estimated to be the best by these first m pulls. In this quantity, $N-n-m$ is the number of pulls that will remain, $\nu_x^{n,m}$ is the expected increment of μ_*^{n+m} over μ_*^n due to the m observations of arm x, and $\nu_x^{n,m} + \mu_*^n$ is the expected value of μ_*^{n+m}.

Compare (8) to the reward obtained without learning by simply pulling the arm with the best mean at time n until the end of the horizon. This reward is $(N-n)\mu_*^n$. Subtracting this from (8) and dividing by m gives the average incremental reward over the m pulls of arm x as

$$\mu_x^n - \mu_*^n + \frac{1}{m}(N-n-m)\,\nu_x^{n,m}. \tag{9}$$

The KG(*) policy finds, for each arm x, the number of pulls $m^*(x)$ that maximizes (9),

$$m^*(x) = \arg \max_{m=1,\ldots,N-n} \mu_x^n - \mu_*^n + \frac{1}{m}(N-n-m)\,\nu_x^{n,m}. \tag{10}$$

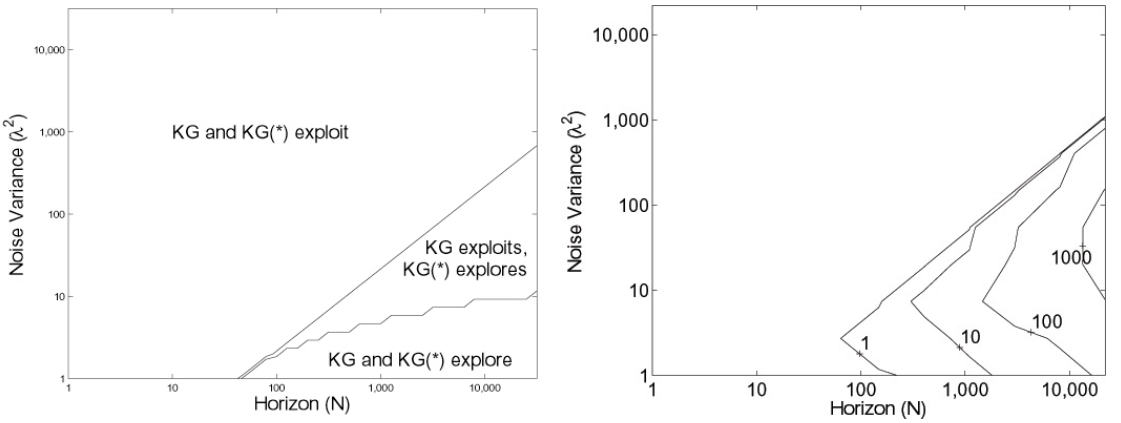
We assign to arm x the resulting maximal reward (a measure of the value of pulling arm x at time n) and choose the arm for which this value is largest. Thus, the KG(*) policy is given by

$$X^{KG,n}(s^n) \in \arg\max_x \max_{m=1,\ldots,N-n} \mu_x^n - \mu_*^n + \frac{1}{m}(N-n-m)\,\nu_x^{n,m}. \tag{11}$$

The KG(*) policy can be viewed as a generalization of KG with m set to $m^*(x)$ rather than 1. By using $m = 1$, the KG policy ignores the sometimes extreme non-concavity of the value of information. The value of one pull can be as low as 10^{-300}, while the value of 10 pulls can be on the order of 10^{-1}. When this occurs, the KG policy is unable to observe the larger values possible by pulling an arm multiple times. To illustrate the differences between the KG and KG(*) policies, we consider the example from above, where one arm has known value 0, and our prior on the other arm is $\mathcal{N}(-1, 1)$, but we vary the horizon and the measurement variance. Figure 2(a) shows the decisions of the two different policies in the first time step, where "exploit" means pulling the known arm (it has the higher mean of the two) and "explore" means pulling the unknown arm. Figure 2(b) shows the difference in expected performance between the two policies, as calculated using Monte Carlo simulation with 1000 independent samples for each variance-horizon point evaluated. The points evaluated were $\{1, e, \ldots, e^{10}\}^2$.

When the horizon is small compared to the noise variance (the "KG and KG(*) exploit" region of Figure 2(a)), learning has relatively little value, and it is not worth making even a single pull of the unknown arm. Consequently, both policies exploit, achieving an objective value of 0 (since the known arm has value 0). In such situations, the KG(*) adjustment is not needed, and the original KG policy works just as well.

When the horizon is large compared to the noise variance (the "KG and KG(*) explore" region of Figure 2(a)), learning has a great deal of value, but this can be discerned from a single arm pull. Therefore, both policies explore on the first measurement (the region at the bottom right of Figure 2(a)). Their resulting difference in value is relatively



(a) First actions taken by KG(*) and KG.

(b) Incremental value of KG(*) over KG.

Figure 2: Performance of KG(*) vs. KG in a two-arm bandit problem where one arm is perfectly known to have value 0, and the other arm has an unknown value with a $\mathcal{N}(-1, 1)$ prior.

small, with the adjustment bringing only a slight advantage to KG(*). In such situations, the KG(*) adjustment may provide some benefit, but this should be weighed against the added complexity of the policy.

It is in the intermediate region, where the horizon and noise variance are large and comparably sized (the "KG exploits, KG(*) explores" region of Figure 2(a)), that the KG(*) adjustment provides value. In this region, unadjusted KG underestimates the value of learning because of the non-concavity of the value of information, and exploits as a result, receiving an expected value of 0. In contrast, the KG(*) adjustment compensates for the lack of concavity and explores. In some cases ($10 \leq \lambda^2 \leq 100$ and $N \geq 20,000$) the KG(*) adjustment obtains an expected reward of more than $1,000$, while without the adjustment the reward is 0. Thus, when the measurement noise is large and the horizon is large enough that exploration may still provide significant value, it is advisable to use KG(*). On the other hand, if the problem does not satisfy these conditions, multiple arm pulls do not substantially add to the information contained in the first pull, which can be interpreted as a kind of robustness of the one-period look-ahead policy.

5. Computational experiments

To compare two policies π_1 and π_2, we take the difference

$$C^{\pi_1,\pi_2} = \sum_{n=0}^{N} \mu_{X^{\pi_1,n}(s^n)} - \mu_{X^{\pi_2,n}(s^n)} \tag{12}$$

of the true mean rewards obtained by running each policy, given the same set of starting data. In order to use this performance measure, it is necessary for the true means μ to be known. Thus, two policies can only be compared in a simulation study, where we can generate many different sets of true means, and then test how well the policies are able to discover those truths.

We generated a set of 100 experimental problems according to the procedure used in the empirical study [27]. Each problem has $M = 100$, with $\sigma_x^0 = 10$ for all x, and every μ_x^0 sampled from the distribution $\mathcal{N}(0, 100)$. The measurement noise was taken to be $\lambda_x^2 = 100$ for all x, and the time horizon was chosen to be $N = 50$. For each problem, we ran each policy under consideration on 10^4 sample paths. In every sample path, a new truth was generated from the prior distribution $\mathcal{N}\left(\mu_x^0, \left(\sigma_x^0\right)^2\right)$ at the beginning of the time horizon. The sample paths were divided into groups of 500, allowing us to obtain approximately normal estimates of the objective value $\mathbf{E}^{\pi} \sum_{n=0}^{N-1} \mu_{X^{\pi,n}(s^n)}$ of each policy π. Taking differences between these values yields estimates of (12).

In addition to KG(*) and KG, we ran five well-known index policies, briefly described below.

Gittins indices (Gitt). We used the approximation of Gittins indices from [13], given by

$$X^{Gitt,n}(s^n) \approx \arg\max_x \mu_x^n + \lambda_x \sqrt{-\log\gamma} \cdot \tilde{b}\left(-\frac{(\sigma_x^n)^2}{\lambda_x^2 \log\gamma}\right)$$

where γ is a discount factor and

$$\tilde{b}(s) = \begin{cases} \frac{s}{\sqrt{2}} & s \leq \frac{1}{7} \\ e^{-0.02645(\log s)^2 + 0.89106 \log s - 0.4873} & \frac{1}{7} < s \leq 100 \\ \sqrt{s}\,(2\log s - \log\log s - \log 16\pi)^{\frac{1}{2}} & s > 100. \end{cases}$$

We considered undiscounted, finite-horizon problems, so γ was treated as a tunable parameter (set to 0.9).

Interval estimation (IE). The interval estimation policy of [7] is given by $X^{IE,n}(s^n) = \arg\max_x \mu_x^n + z \cdot \sigma_x^n$, where z is a tunable parameter. We found that $z = 1.5$ worked well for the problems we generated. When tuned properly, IE gave the best performance aside from the KG variants, but proved to be highly sensitive to the choice of z.

Upper confidence bound (UCB). The UCB policy of [9] is given by

$$X^{UCB,n}(s^n) = \mu_x^n + \sqrt{\frac{2}{N_x^n} g\left(\frac{N_x^n}{N}\right)}$$

where N_x^n is the number of times we have pulled arm x up to and including time n, and

$$g(t) = \log\frac{1}{t} - \frac{1}{2}\log\log\frac{1}{t} - \frac{1}{2}\log 16\pi.$$

It has been shown by [8, 9] that the number of times that any suboptimal arm is pulled under this policy is $O(\log N)$.

Epsilon-greedy (Eps). The epsilon-greedy policy (see e.g. [28]) chooses an arm at random with probability $\frac{1}{n}$ and pulls the arm given by $\arg\max_x \mu_x^n$ the rest of the time.

Pure exploitation (Exp). The pure exploitation policy is given by $X^{Exp,n}(s^n) = \arg\max_x \mu_x^n$.

5.1. Comparison of KG(*) to KG

Table 1 gives the mean values of (12) across 100 problems with KG(*) as π_1 and the other policies as π_2. The standard errors of these values are also given in the table. Figure 3 shows the distribution of the sampled values of (12). Each histogram is labeled with the two policies being compared, and indicates how often KG(*) outperformed the competition. Bars to the right of zero indicate problems where KG(*) outperformed another policy, and bars to the left of zero indicate the opposite. We see that KG(*) consistently outperforms all the index policies in the comparison. Only interval estimation is ever able to outperform KG(*). However, this only occurs 30% of the time, and only one of those times is statistically significant. KG(*) always outperforms the other four index policies by a statistically significant margin.

By contrast, KG(*) achieves only a small improvement over regular KG. On 93 problems, the difference between KG(*) and KG is not statistically significant. Of the remaining seven problems, KG(*) outperforms KG four times, and is outperformed three times. Essentially, the two policies are comparable. For the given set of problems, we do not derive much additional benefit from looking out more than one step when we make decisions. Most of the valuable information can be gleaned by a one-period look-ahead. In the language of Section 4, these problems fall into the large region where KG is robust, and the adjustment adds only incremental value. However, there are other problem settings (related to high values of λ_x^2 and N) for which the adjustment adds more significant improvement.

KG(*) vs.	KG	Gitt	IE	UCB	Eps	Exp
Mean	0.0906	54.7003	4.5986	1522.7409	526.3927	375.1965
SE	6.8715	7.4696	6.9800	25.0439	9.3499	9.7285

Table 1: Means and standard errors for the experiments.

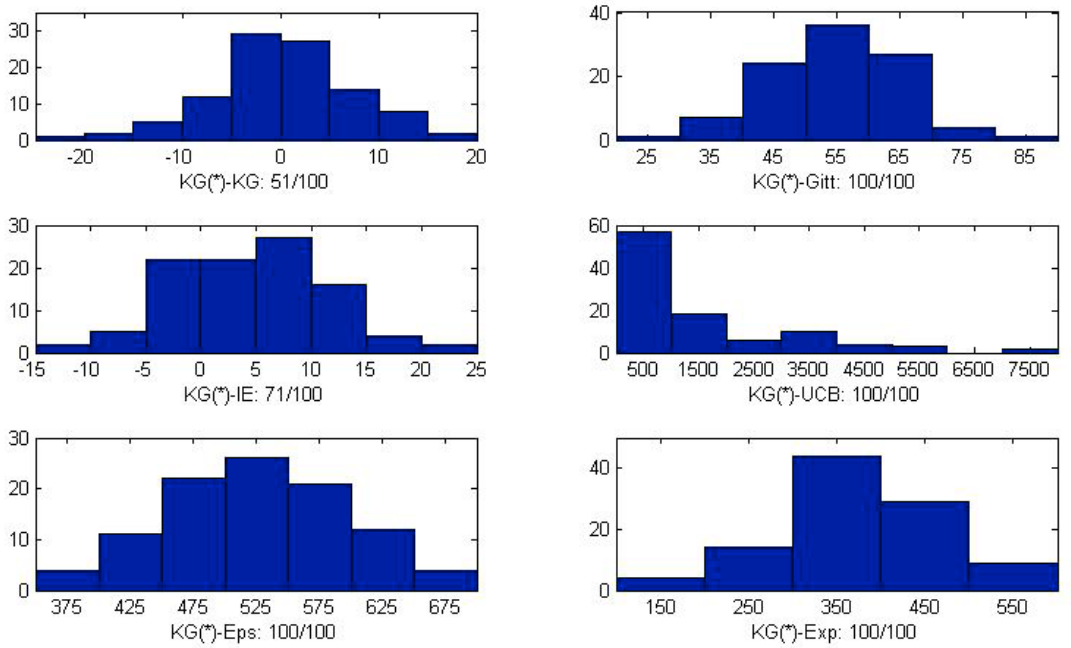


Figure 3: Histograms showing the performance of KG(*) relative to other policies across 100 test problems.

5.2. *Effect of measurement noise and time horizon on the comparison*

We considered the effect of λ_x^2 and N on the performance of KG(*) relative to KG. The value of information tends to be heavily non-concave when at least one of these parameters is large. We examined the performance of KG(*), KG and the two best-performing index policies on one problem chosen at random from our set. In Figure 4(a), the measurement noise λ_x^2 is varied in the same way across all x relative to the magnitude of $\left(\sigma_x^0\right)^2 = 100$. In Figure 4(b), the time horizon N is varied, while the measurement noise is fixed at the baseline value of $\lambda^2 = 100$.

Predictably, the suboptimality of each policy increases in each case. What is interesting, however, is that for large enough noise, the suboptimality no longer seems to depend on the policy. The KG(*) policy maintains a slight advantage over other policies for $\lambda^2 \geq 100$, but this difference is not statistically significant. On the other hand, for N large enough, KG(*) pulls ahead of KG, and this difference only increases with the time horizon.

These results confirm the insight of our small example from Figure 2(a). If we fix N and increase the measurement noise in that example, we move into the region where both policies make the same decision. It is only when we increase N for a fixed value of the measurement noise that KG(*) brings about a significant improvement. The same tendencies can be seen in our larger experimental study.

It should be noted that we only start to see an improvement in Figure 4(b) once N is on the order of 10^3. In most of our motivating examples, the time horizon would be much smaller, and it would be sufficient to use the KG policy. However, if we are dealing with an infinite-horizon problem, the non-concavity of the value of information becomes a much more serious concern.

6. Conclusion

We have analyzed the question of whether a one-period look-ahead policy is able to collect enough information to make good decisions in multi-armed bandit problems. We compared the one-period look-ahead to an adjusted policy that approximates a multi-step look-ahead. In a large data set, the adjusted policy has a slight advantage over the

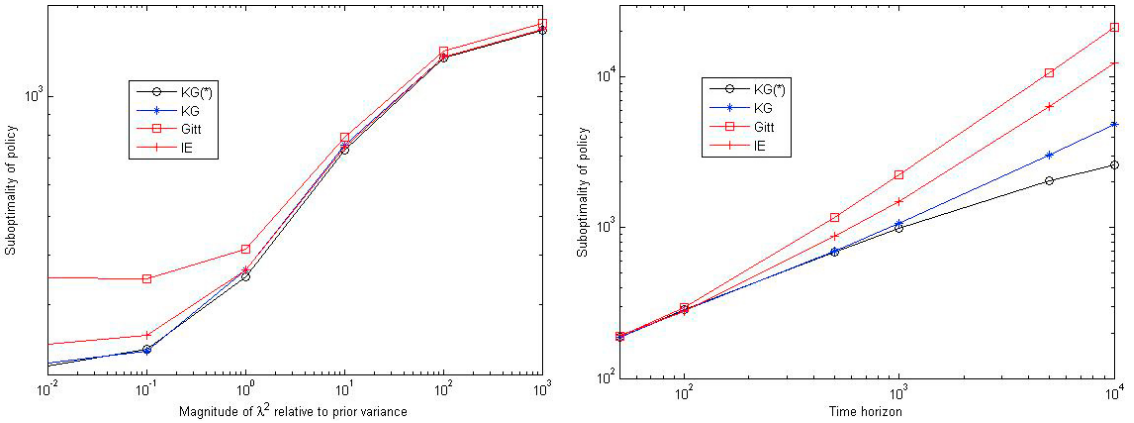


Figure 4: Effect of (a) the measurement noise λ^2 and (b) the time horizon N on the performance of KG(*) and other policies.

one-period look-ahead on average, but the difference tends to be insignificant. However, the difference can become more compelling in special cases where the time horizon is large enough.

In many settings, most of the useful information about an arm can be obtained in a single pull. Because the KG policy considers our beliefs about all arms when computing the value of one pull, the KG logic is able to capture much of the complexity of the problem. The KG policy often does not need to be adjusted to handle multiple measurements, though adjustment may improve performance in certain specific problem classes (those involving a large time horizon). Because an adjusted multi-step look-ahead frequently yields only incremental improvement over a policy that only considers one time step, we conclude that a one-period look-ahead policy is a robust approach to finite-horizon bandit problems.

Acknowledgments

This work was supported in part by AFOSR contract FA9550-08-1-0195 and ONR contract N00014-07-1-0150 through the Center for Dynamic Data Analysis.

References

[1] J. C. Gittins, D. M. Jones, A dynamic allocation index for the discounted multiarmed bandit problem, Biometrika 66 (3) (1979) 561–565.
[2] D. Berry, L. Pearson, Optimal designs for clinical trials with dichotomous responses, Statistics in Medicine 4 (4) (1985) 497–508.
[3] M. Babaioff, Y. Sharma, A. Slivkins, Characterizing truthful multi-armed bandit mechanisms, in: Proceedings of the 10th ACM Conference on Electronic Commerce, 2009, pp. 79–88.
[4] I. O. Ryzhov, W. B. Powell, A Monte Carlo Knowledge Gradient Method For Learning Abatement Potential Of Emissions Reduction Technologies, in: M. Rosetti, R. Hill, B. Johansson, A. Dunkin, R. Ingalls (Eds.), Proceedings of the 2009 Winter Simulation Conference, 2009, pp. 1492–1502.
[5] D. A. Berry, B. Fristedt, Bandit Problems, Chapman and Hall, London, 1985.
[6] J. Gittins, Multi-Armed Bandit Allocation Indices, John Wiley and Sons, New York, 1989.
[7] L. P. Kaelbling, Learning in Embedded Systems, MIT Press, Cambridge, MA, 1993.
[8] T. L. Lai, H. Robbins, Asymptotically efficient adaptive allocation rules, Advances in Applied Mathematics 6 (1985) 4–22.
[9] T. Lai, Adaptive treatment allocation and the multi-armed bandit problem, The Annals of Statistics 15 (3) (1987) 1091–1114.
[10] J. C. Gittins, D. M. Jones, A dynamic allocation index for the sequential design of experiments, in: J. Gani (Ed.), Progress in Statistics, 1974, pp. 241–266.
[11] M. Brezzi, T. Lai, Optimal learning and experimentation in bandit problems, Journal of Economic Dynamics and Control 27 (1) (2002) 87–108.
[12] Y. Yao, Some results on the Gittins index for a normal reward process, in: H. Ho, C. Ing, T. Lai (Eds.), Time Series and Related Topics: In Memory of Ching-Zong Wei, Institute of Mathematical Statistics, Beachwood, OH, USA, 2006, pp. 284–294.
[13] S. Chick, N. Gans, Economic analysis of simulation selection options, Management Science 55 (3) (2009) 421–437.

[14] S. Gupta, K. Miescke, Bayesian look ahead one-stage sampling allocations for selection of the best population, Journal of statistical planning and inference 54 (2) (1996) 229–244.
[15] P. I. Frazier, W. B. Powell, S. Dayanik, A knowledge gradient policy for sequential information collection, SIAM Journal on Control and Optimization 47 (5) (2008) 2410–2439.
[16] P. I. Frazier, W. B. Powell, S. Dayanik, The knowledge-gradient policy for correlated normal rewards, INFORMS J. on Computing 21 (4) (2009) 599–613.
[17] S. Chick, J. Branke, C. Schmidt, Sequential Sampling to Myopically Maximize the Expected Value of Information, INFORMS J. on Computing (to appear).
[18] I. O. Ryzhov, W. B. Powell, P. I. Frazier, The knowledge gradient algorithm for a general class of online learning problems, Submitted for publication.
[19] I. O. Ryzhov, W. B. Powell, The knowledge gradient algorithm for online subset selection, in: Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, Nashville, TN, 2009, pp. 137–144.
[20] P. I. Frazier, W. B. Powell, Paradoxes in Learning: The Marginal Value of Information and the Problem of Too Many Choices, Submitted for publication.
[21] M. H. DeGroot, Optimal Statistical Decisions, John Wiley and Sons, 1970.
[22] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, Machine Learning 47 (2-3) (2002) 235–256.
[23] R. Howard, Information value theory, IEEE Transactions on systems science and cybernetics 2 (1) (1966) 22–26.
[24] H. Chade, E. Schlee, Another look at the Radner–Stiglitz nonconcavity in the value of information, Journal of Economic Theory 107 (2) (2002) 421–452.
[25] R. Radner, J. Stiglitz, A Nonconcavity in the Value of Information, Bayesian models in economic theory 5 (1984) 33–52.
[26] S. E. Chick, P. I. Frazier, The Conjunction Of The Knowledge Gradient And The Economic Approach To Simulation Selection, in: M. Rosetti, R. Hill, B. Johansson, A. Dunkin, R. Ingalls (Eds.), Proceedings of the 2009 Winter Simulation Conference, 2009, pp. 528–539.
[27] J. Vermorel, M. Mohri, Multi-armed bandit algorithms and empirical evaluation, Proceedings of the 16th European Conference on Machine Learning (2005) 437–448.
[28] R. Sutton, A. Barto, Reinforcement Learning, The MIT Press, Cambridge, Massachusetts, 1998.